Perform EVAT GenerativeAI multi-table generation comparison

Load the Original Data Model tables

Define the original data model entity relationships

Train the AI Model on the above data.

Generate a new Dataset using AI synthesis.

Compare the original data Model Versus the synthesized dataset/

Data preparation

Fetch & Review original Data Model Files

```
# prompt: load  ChargingSession.csv into a dataframe ChargingSession, load OrderScheduling.csv into a dataframe OrderScheduling, loa

import pandas as pd

# Load the datasets into pandas DataFrames
ChargingSession = pd.read_csv('ChargingSession.csv')
OrderScheduling = pd.read_csv('OrderScheduling.csv')
PurchaseWholesaleElec = pd.read_csv('PurchaseWholesaleElec.csv')
Stations = pd.read_csv('Stations.csv')
UserPayment = pd.read_csv('UserPayment.csv')
Users = pd.read_csv('Users.csv')
ev = pd.read_csv('ev.csv')
```

```
ChargingSession.head(2)
```

	Id	SlotID	UserId	StationId	LocationId	ChargeStartTime	ChargeEndTime	ChargeTimeHrs	KwhTotal	CustPricePerKW	Cu
0	4926737	841344	78908148	730023	878393	7/10/2023 7:18	7/10/2023 8:44	0 days 01:26:00	82.049511	0.570110	
1	3738844	252642	78908148	730023	878393	5/08/2023 5:29	5/08/2023 7:01	0 days 01:32:00	88.246562	0.532488	

Next steps: Generate code with ChargingSession | View recommended plots | New interactive sheet

```
OrderScheduling.head(2)
```

	Id	UserId	StationId	SlotId	ConnectorType	BookingDate	SlotStartTime	SlotLength	EstCost50KWCharger	BookingTimeS
0	4926737	78908148	730023	841344	CCS	7/10/2023	7	2	55.0	[7,
1	3738844	78908148	730023	252642	CCS	5/08/2023	5	3	82.5	[5, 6,

```
PurchaseWholesaleElec.head(2)
```

	Id	StationId	SupplyId	ExpenditureCode	RRPperKW	RRPperKW24hrs	BasePrice	PeakPriceAdjustment	CustPricePerKW	Purch
0	4926737	730023	VIC	Wholesale_cost	0.01156	0.020110	0.55	0	0.570110	7/
1	3738844	730023	VIC	Wholesale_cost	0.00895	-0.017512	0.55	0	0.532488	5/

```
Stations.head(2)
```

	Id	ChargingPoints	Latitude	Longitude	OperatorID	Address	Amenities	BasePrice	PeakPriceAdjustment	EstCost50KWCh
0	488364	2	-37.813437	144.955934	TBD	535 Little Lonsdale St, Melbourne VIC 3004, Au...	Grocery, Valet Parking, Coffee	0.55	0	
1	540225	10	-37.818758	144.958866	TBD	450 Flinders Ln, Melbourne VIC 3000, Australia	Dining, Grocery, Restrooms, Car Cleaning, Coff...	0.55	0	

```
UserPayment.head(2)
```

	Id	BookingDeposit	PurchaseId	PaymentType	CustPayAmount	TransAmount	UserId	StationId	PaymentDate	ChargingId	O
0	4926737	1	4926737	MasterCard	46.777252	45.777252	78908148	730023	7/10/2023 8:44	4926737	4
1	3738844	1	3738844	Visa	46.990274	45.990274	78908148	730023	5/08/2023 7:01	3738844	3

```
Users.head(2)
```

	Id	UserFirstName	UserSurname	UserFullName	UserEmailAddress	UserPassword	UserRole	UserHomeAddress	UserMo
0	78908148	Leslie	Shelton	Jame Martinez	mileage1881@duck.com	U]s$k]Xr	user	340 Larkstone Bridge	
1	35897499	Bob	Bender	Branda Delacruz	interpreted1898@yandex.com	5WQcKdpb	user	389 Leenagrenagh Creek	

```
ev.head(2)
```

	Id	Manufacturer	Model	VehicleYear	VehicleClass	FuelType	ConnectorType	BatteryCapacityKwh	EstBattChargeTime80pe
0	IJWO2541	Audi	Audi e-tron	2024	Large SUV	BEV	CCS	106.488	89
1	JQVH4255	Audi	Audi e-tron GT	2024	Large Car	BEV	CCS	103.680	87

```
## Train a multi-table generator

Configuring a mutli-table generator is simply done, by configuring each table,
their primary key as well as all their foreign key relations.
```

Train a multi-table generator

Configuring a mutli-table generator is simply done, by configuring each table, their primary key as well as all their foreign key relations.

```
!pip install -U mostlyai
```

```
Collecting mostlyai
  Downloading mostlyai-0.7.0-py3-none-any.whl.metadata (5.4 kB)
Collecting httpx<0.28.0,>=0.25.0 (from mostlyai)
  Downloading httpx-0.27.2-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: pandas<3.0.0,>=1.5.3 in /usr/local/lib/python3.10/dist-packages (from mostlyai) (2.2.2)
Requirement already satisfied: pyarrow>=14.0.0 in /usr/local/lib/python3.10/dist-packages (from mostlyai) (17.0.0)
Requirement already satisfied: pydantic<3.0.0,>=2.4.2 in /usr/local/lib/python3.10/dist-packages (from mostlyai) (2.10.4)
Requirement already satisfied: rich>=13.7.0 in /usr/local/lib/python3.10/dist-packages (from mostlyai) (13.9.4)
Requirement already satisfied: anyio in /usr/local/lib/python3.10/dist-packages (from httpx<0.28.0,>=0.25.0->mostlyai) (3.7.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx<0.28.0,>=0.25.0->mostlyai) (2024.1
```

```
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.10/dist-packages (from httpx<0.28.0,>=0.25.0->mostlyai) (
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from httpx<0.28.0,>=0.25.0->mostlyai) (3.10)
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx<0.28.0,>=0.25.0->mostlyai) (1.3.1)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.10/dist-packages (from httpcore==1.*->httpx<0.28.0,>=0
Requirement already satisfied: numpy>=1.22.4 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0.0,>=1.5.3->mostlyai) (1
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0.0,>=1.5.3->mos
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0.0,>=1.5.3->mostlyai) (20
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas<3.0.0,>=1.5.3->mostlyai) (
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3.0.0,>=2.4.2->m
Requirement already satisfied: pydantic-core==2.27.2 in /usr/local/lib/python3.10/dist-packages (from pydantic<3.0.0,>=2.4.2->mo
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.10/dist-packages (from pydantic<3.0.0,>=2.4.2
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=13.7.0->mostlyai) (3
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=13.7.0->mostlyai)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.0->rich>=13.7.0->
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas<3.0.0,>=
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx<0.28.0,>=0.25.0->mos
Downloading mostlyai-0.7.0-py3-none-any.whl (45 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.0/46.0 kB 2.0 MB/s eta 0:00:00
Downloading httpx-0.27.2-py3-none-any.whl (76 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.4/76.4 kB 4.1 MB/s eta 0:00:00
Installing collected packages: httpx, mostlyai
  Attempting uninstall: httpx
    Found existing installation: httpx 0.28.1
    Uninstalling httpx-0.28.1:
      Successfully uninstalled httpx-0.28.1
Successfully installed httpx-0.27.2 mostlyai-0.7.0
```

```
from mostlyai import MostlyAI

# initialize client
mostly = MostlyAI(api_key='mostly-a85530a98a3abb708825e71e667bb3df2350fbaa90eb7ba4fccbb84e0a07cf53')
```

Connected to https://app.mostly.ai (v321) as **john.collins@sensation360.com**

```
# Define the configuration for the generator
config = {
    "name": "Multi-table - EVAT2",
    "tables": [
        {
            "name": "ev",
            "primary_key": "Id",
            "columns": [
                {"name": "Id", "type": "integer"},
                {"name": "Manufacturer", "type": "string"},
                {"name": "Model", "type": "string"},
                {"name": "VehicleYear", "type": "integer"},
                {"name": "VehicleClass", "type": "string"},
                {"name": "FuelType", "type": "string"},
                {"name": "ConnectorType", "type": "string"},
                {"name": "BatteryCapacityKwh", "type": "float"},
                {"name": "EstBattChargeTime80perc50KW", "type": "float"},
                {"name": "EnergyConsumptionWhkm", "type": "float"},
                {"name": "ElectricRangeKm", "type": "float"},
                {"name": "UserId", "type": "integer"}
            ],
            "data": ev,
        },
        {
            "name": "Users",
            "primary_key": "Id",
            "columns": [
                {"name": "Id", "type": "integer"},
                {"name": "UserFirstName", "type": "string"},
                {"name": "UserSurname", "type": "string"},
                {"name": "UserFullName", "type": "string"},
                {"name": "UserEmailAddress", "type": "string"},
                {"name": "UserPassword", "type": "string"},
                {"name": "UserRole", "type": "string"},
                {"name": "UserHomeAddress", "type": "string"},
                {"name": "UserMobilePhoneNumber", "type": "integer"},
                {"name": "UserAuthenticated", "type": "integer"},
                {"name": "VehicleID", "type": "integer"}
            ],
            "foreign_keys": [
                {"column": "VehicleID", "references": "ev.Id", "referencedTable": "ev", "is_context": True} # Added referencedTable
            ],
            "data": Users,
        },
```

```
        {
            "name": "Stations",
            "primary_key": "Id",
            "columns": [
                {"name": "Id", "type": "integer"},
                {"name": "ChargingPoints", "type": "integer"},
                {"name": "Latitude", "type": "float"},
                {"name": "Longitude", "type": "float"},
                {"name": "OperatorID", "type": "string"},
                {"name": "Address", "type": "string"},
                {"name": "Amenities", "type": "string"},
                {"name": "BasePrice", "type": "float"},
                {"name": "PeakPriceAdjustment", "type": "float"},
                {"name": "EstCost50KWChargerHr", "type": "float"},
                {"name": "SlotIDs", "type": "string"},
                {"name": "ConnectionTypes", "type": "string"},
                {"name": "PaymentTypes", "type": "string"},
                {"name": "PowerOutput", "type": "float"},
                {"name": "LocationName", "type": "string"},
                {"name": "PostalCode", "type": "integer"},
                {"name": "LocationID", "type": "integer"}
            ],
            "data": Stations,
        },
        {
            "name": "OrderScheduling",
            "primary_key": "Id",
            "columns": [
                {"name": "Id", "type": "integer"},
                {"name": "UserId", "type": "integer"},
                {"name": "StationId", "type": "integer"},
                {"name": "SlotId", "type": "integer"},
                {"name": "ConnectorType", "type": "string"},
                {"name": "BookingDate", "type": "datetime"},
                {"name": "SlotStartTime", "type": "integer"},
                {"name": "SlotLength", "type": "integer"},
                {"name": "EstCost50KWCharger", "type": "float"},
                {"name": "BookingTimeSlots", "type": "string"},
                {"name": "BookingDeposit", "type": "float"}
            ],
            "foreign_keys": [
                {"column": "UserId", "references": "Users.Id", "referencedTable": "Users", "is_context": False}, # Added referencedT
                {"column": "StationId", "references": "Stations.Id", "referencedTable": "Stations", "is_context": True} # Added refe
            ],
            "data": OrderScheduling
        },
                {
            "name": "UserPayment",
            "primary_key": "Id",
            "columns": [
                {"name": "Id", "type": "integer"},
                {"name": "BookingDeposit", "type": "float"},
                {"name": "PurchaseId", "type": "integer"},
                {"name": "PaymentType", "type": "string"},
                {"name": "CustPayAmount", "type": "float"},
                {"name": "TransAmount", "type": "float"},
                {"name": "UserId", "type": "integer"},
                {"name": "StationId", "type": "integer"},
                {"name": "PaymentDate", "type": "datetime"},
                {"name": "ChargingId", "type": "integer"},
                {"name": "OrderId", "type": "integer"}
            ],
            "foreign_keys": [
                {"column": "PurchaseId", "references": "PurchaseWholesaleElec.Id", "referencedTable": "PurchaseWholesaleElec", "is_c
                {"column": "StationId", "references": "Stations.Id", "referencedTable": "Stations", "is_context": False}, # Added re
                {"column": "UserId", "references": "Users.Id", "referencedTable": "Users", "is_context": False}, # Added referencedT
                {"column": "OrderId", "references": "OrderScheduling.Id", "referencedTable": "OrderScheduling", "is_context": False}
                {"column": "ChargingId", "references": "ChargingSession.Id", "referencedTable": "ChargingSession", "is_context": Tru
            ],
            "data": UserPayment, # Use the UserPayment DataFrame directly
        },
        {
            "name": "PurchaseWholesaleElec",
            "primary_key": "Id",
            "columns": [
                {"name": "Id", "type": "integer"},
                {"name": "StationId", "type": "integer"},
                {"name": "SupplyId", "type": "string"},
                {"name": "ExpenditureCode", "type": "string"},
                {"name": "RRPperKW", "type": "float"},
```

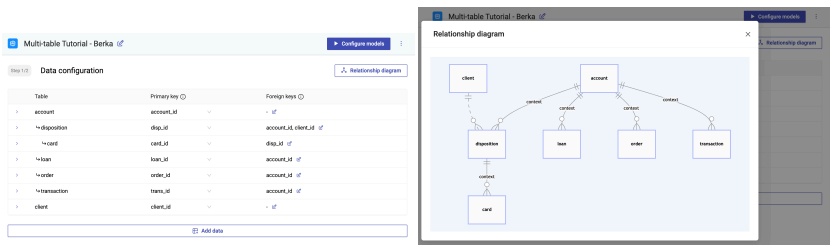
```
                {"name": "RRPperKW24hrs", "type": "float"},
                {"name": "BasePrice", "type": "float"},
                {"name": "PeakPriceAdjustment", "type": "float"},
                {"name": "CustPricePerKW", "type": "float"},
                {"name": "PurchaseDate", "type": "datetime"}
            ],
            "foreign_keys": [
                {"column": "StationId", "references": "Stations.Id", "referencedTable": "Stations", "is_context": True} # Added refe
            ],
            "data": PurchaseWholesaleElec,
        },
        {
            "name": "ChargingSession",
            "primary_key": "Id",
            "columns": [
                {"name": "Id", "type": "integer"},
                {"name": "SlotID", "type": "integer"},
                {"name": "UserId", "type": "integer"},
                {"name": "StationId", "type": "integer"},
                {"name": "LocationId", "type": "integer"},
                {"name": "ChargeStartTime", "type": "datetime"},
                {"name": "ChargeEndTime", "type": "datetime"},
                {"name": "ChargeTimeHrs", "type": "string"},
                {"name": "KwhTotal", "type": "float"},
                {"name": "CustPricePerKW", "type": "float"},
                {"name": "CustPayAmount", "type": "float"},
                {"name": "BookingDate", "type": "datetime"},
                {"name": "ConnectorType", "type": "string"},
                {"name": "OrderId", "type": "integer"},
                {"name": "PurchaseId", "type": "integer"}
            ],
            "foreign_keys": [
                {"column": "OrderId", "references": "OrderScheduling.Id", "referencedTable": "OrderScheduling", "is_context": True},
                {"column": "UserId", "references": "Users.Id", "referencedTable": "Users", "is_context": False}, # Added referencedT
                {"column": "StationId", "references": "Stations.Id", "referencedTable": "Stations", "is_context": False}, # Added re
                {"column": "PurchaseId", "references": "PurchaseWholesaleElec.Id", "referencedTable": "PurchaseWholesaleElec", "is_c
            ],
            "data": ChargingSession
        }
    ]
}

# Display the generator configuration
#config
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

```
# configure a generator, but don't yet start the training thereof
g = mostly.train(config=config, start=False)

# open generator in a new browser tab
g.open()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
Created generator 917375b2-3456-4feb-bd10-31af72ea82bf
'https://app.mostly.ai/d/generators/917375b2-3456-4feb-bd10-31af72ea82bf'
```

You can now also inspect the configuration on the Web UI. It will look like this:



Now, launch the training, and wait for it to be finished. This shouldn't take longer than 10 minutes.

```
g.training.start()
g = g.training.wait(progress_bar=True)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
Overall job progress                                                    100% 0:01:02
Step ev:tabular PULL_TRAINING_DATA                                      100% 0:00:05
Step ev:tabular ANALYZE_TRAINING_DATA                                   100% 0:00:05
Step ev:tabular ENCODE_TRAINING_DATA                                    100% 0:00:02
Step ev:tabular TRAIN_MODEL 💎                                          100% 0:00:02
Step ev:tabular GENERATE_MODEL_REPORT_DATA                              100% 0:00:00
Step ev:tabular CREATE_MODEL_REPORT                                     100% 0:00:00
Step Users:tabular PULL_TRAINING_DATA                                   100% 0:00:02
Step Users:tabular ANALYZE_TRAINING_DATA                                100% 0:00:05
Step Users:tabular ENCODE_TRAINING_DATA                                 100% 0:00:02
Step Users:tabular TRAIN_MODEL 💎                                       100% 0:00:02
Step Users:tabular GENERATE_MODEL_REPORT_DATA                           100% 0:00:00
Step Users:tabular CREATE_MODEL_REPORT                                  100% 0:00:00
Step Stations:tabular PULL_TRAINING_DATA                                100% 0:00:05
Step Stations:tabular ANALYZE_TRAINING_DATA                             100% 0:00:02
Step Stations:tabular ENCODE_TRAINING_DATA                              100% 0:00:02
Step Stations:tabular TRAIN_MODEL 💎                                    100% 0:00:02
Step Stations:tabular GENERATE_MODEL_REPORT_DATA                        100% 0:00:00
Step Stations:tabular CREATE_MODEL_REPORT                               100% 0:00:10
Step OrderScheduling:tabular PULL_TRAINING_DATA                         100% 0:00:02
Step OrderScheduling:tabular ANALYZE_TRAINING_DATA                      100% 0:00:05
Step OrderScheduling:tabular ENCODE_TRAINING_DATA                       100% 0:00:02
Step OrderScheduling:tabular TRAIN_MODEL 💎                             100% 0:00:05
Step OrderScheduling:tabular GENERATE_MODEL_REPORT_DATA                 100% 0:00:02
Step OrderScheduling:tabular CREATE_MODEL_REPORT                        100% 0:00:00
Step ChargingSession:tabular PULL_TRAINING_DATA                         100% 0:00:05
Step ChargingSession:tabular ANALYZE_TRAINING_DATA                      100% 0:00:05
Step ChargingSession:tabular ENCODE_TRAINING_DATA                       100% 0:00:05
Step ChargingSession:tabular TRAIN_MODEL 💎                             100% 0:00:13
Step ChargingSession:tabular GENERATE_MODEL_REPORT_DATA                 100% 0:00:00
Step ChargingSession:tabular CREATE_MODEL_REPORT                        100% 0:00:13
Step UserPayment:tabular PULL_TRAINING_DATA                             100% 0:00:05
Step UserPayment:tabular ANALYZE_TRAINING_DATA                          100% 0:00:05
Step UserPayment:tabular ENCODE_TRAINING_DATA                           100% 0:00:05
Step UserPayment:tabular TRAIN_MODEL 💎                                 100% 0:00:13
Step UserPayment:tabular GENERATE_MODEL_REPORT_DATA                     100% 0:00:02
Step UserPayment:tabular CREATE_MODEL_REPORT                            100% 0:00:13
Step PurchaseWholesaleElec:tabular PULL_TRAINING_DATA                   100% 0:00:05
Step PurchaseWholesaleElec:tabular ANALYZE_TRAINING_DATA                100% 0:00:05
Step PurchaseWholesaleElec:tabular ENCODE_TRAINING_DATA                 100% 0:00:02
Step PurchaseWholesaleElec:tabular TRAIN_MODEL 💎                       100% 0:00:05
Step PurchaseWholesaleElec:tabular GENERATE_MODEL_REPORT_DATA           100% 0:00:02
```

Generate a multi-table dataset

Once, the training has completed, you can generate a multi-table dataset with. If you do not specify the sample size, then the platform will generate as many subject records, as there were in the original subject tables. Otherwise, you will need to specify for each subject table, the number of records, as these are independently sampled.

```
# use generator to create a synthetic dataset
sd = mostly.generate('917375b2-3456-4feb-bd10-31af72ea82bf', start=False)
sd.open()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
Created synthetic dataset 66523b04-edb4-4c86-a504-483bbf548c4a with generator 92b02a05-3696-4667-b0aa-1ec3f575d61c
'https://app.mostly.ai/d/synthetic-datasets/66523b04-edb4-4c86-a504-483bbf548c4a'
```

```
sd.generation.start()
sd = sd.generation.wait(progress_bar=True)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
Overall job progress                                  88% 0:02:02
Step Stations:tabular GENERATE_DATA                  100% 0:00:00
Step Stations:tabular CREATE_DATA_REPORT             100% 0:00:07
Step OrderScheduling:tabular GENERATE_DATA           100% 0:00:07
Step OrderScheduling:tabular CREATE_DATA_REPORT      100% 0:00:00
Step PurchaseWholesaleElec:tabular GENERATE_DATA     100% 0:00:07
Step PurchaseWholesaleElec:tabular CREATE_DATA_REPORT 100% 0:00:00
Step ChargingSession:tabular GENERATE_DATA           100% 0:00:15
Step ChargingSession:tabular CREATE_DATA_REPORT      100% 0:00:10
Step UserPayment:tabular GENERATE_DATA               100% 0:00:25
Step UserPayment:tabular CREATE_DATA_REPORT          100% 0:00:07
Step ev:tabular GENERATE_DATA                        100% 0:00:00
Step ev:tabular CREATE_DATA_REPORT                   100% 0:00:00
Step Users:tabular GENERATE_DATA                     100% 0:00:05
Step Users:tabular CREATE_DATA_REPORT                100% 0:00:00
Step common FINALIZE_GENERATION                        0% 0:00:04
Step common DELIVER_DATA                               0% -:--:--
Step  FINALIZE_GENERATION failed
```

Fetch the Generated Data

```
# fetch configuration via API
sd = mostly.synthetic_datasets.get('54b2599b-fbc2-44f6-ba0f-d6707fa74ee3')
config = sd.config()
config
```

```
 'name': 'EVAT_DB',
 'description': 'EVAT Charging Session Data',
 'tables': [
     {
         'name': 'ev',
         'configuration': {
             'sample_size': 85,
             'sample_seed_connector_id': None,
             'sample_seed_dict': None,
             'sample_seed_data': None,
             'sampling_temperature': 1.0,
             'sampling_top_p': 1.0,
             'rebalancing': None,
             'imputation': None,
             'fairness': None,
             'tabular_compute': 'c5f0d5da-04d9-4099-8394-e1048a469a5a',
             'language_compute': None
         }
     },
     {
         'name': 'Users',
         'configuration': {
             'sample_size': None,
             'sample_seed_connector_id': None,
             'sample_seed_dict': None,
             'sample_seed_data': None,
             'sampling_temperature': 1.0,
             'sampling_top_p': 1.0,
             'rebalancing': None,
             'imputation': None,
             'fairness': None,
             'tabular_compute': 'c5f0d5da-04d9-4099-8394-e1048a469a5a',
             'language_compute': None
         }
     },
     {
         'name': 'Stations',
         'configuration': {
             'sample_size': 105,
             'sample_seed_connector_id': None,
             'sample_seed_dict': None,
             'sample_seed_data': None,
             'sampling_temperature': 1.0,
             'sampling_top_p': 1.0,
             'rebalancing': None,
             'imputation': None,
             'fairness': None,
             'tabular_compute': 'c5f0d5da-04d9-4099-8394-e1048a469a5a',
             'language_compute': 'd2dc8ce3-2861-4bea-95c2-99c01a2ed084'
         }
     },
     {
         'name': 'OrderScheduling',
         'configuration': {
             'sample_size': None,
             'sample_seed_connector_id': None,
             'sample_seed_dict': None,
             'sample_seed_data': None,
             'sampling_temperature': 1.0,
             'sampling_top_p': 1.0,
             'rebalancing': None,
             'imputation': None,
             'fairness': None,
             'tabular_compute': 'c5f0d5da-04d9-4099-8394-e1048a469a5a',
             'language_compute': None
         }
     },
     {
         'name': 'ChargingSession',
         'configuration': {
             'sample_size': None,
             'sample_seed_connector_id': None,
             'sample_seed_dict': None,
             'sample_seed_data': None,
             'sampling_temperature': 1.0,
             'sampling_top_p': 1.0,
             'rebalancing': None,
             'imputation': None,
             'fairness': None,
             'tabular_compute': 'c5f0d5da-04d9-4099-8394-e1048a469a5a',
             'language_compute': None
         }
     },
     {
         'name': 'UserPayment',
```

```
            'configuration': {
                'sample_size': None,
                'sample_seed_connector_id': None,
                'sample_seed_dict': None,
                'sample_seed_data': None,
                'sampling_temperature': 1.0,
                'sampling_top_p': 1.0,
                'rebalancing': None,
                'imputation': None,
                'fairness': None,
                'tabular_compute': 'c5f0d5da-04d9-4099-8394-e1048a469a5a',
                'language_compute': None
            }
        },
        {
            'name': 'PurchaseWholesaleElec',
            'configuration': {
                'sample_size': None,
                'sample_seed_connector_id': None,
                'sample_seed_dict': None,
                'sample_seed_data': None,
                'sampling_temperature': 1.0,
                'sampling_top_p': 1.0,
                'rebalancing': None,
                'imputation': None,
                'fairness': None,
                'tabular_compute': 'c5f0d5da-04d9-4099-8394-e1048a469a5a',
                'language_compute': None
            }
        }
    ]
```

```
# once done, fetch the synthetic data as dictionary of DataFrames
synthetics = sd.data()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

Show sample records for each table

```
for k in synthetics:
    print("===", k, "===")
    display(synthetics[k].sample(n=3))
```

```
=== Users ===
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

	Id	UserFirstName	UserSurname	UserFullName	UserEmailAddress	UserPassword	UserRole	UserHomeAddress	UserMobil
31	mostlyff-aa94-406b-862c-17dd5d1332ca	_RARE_	_RARE_	_RARE_	_RARE_	_RARE_	user	_RARE_	
28	mostly13-a658-4b5a-a0c5-ef6a4bf964be	_RARE_	_RARE_	_RARE_	_RARE_	_RARE_	user	_RARE_	
54	mostly57-9bc7-4d41-852e-c0c8a82163cb	_RARE_	_RARE_	_RARE_	_RARE_	_RARE_	user	_RARE_	

```
=== OrderScheduling ===
```

	Id	UserId	StationId	SlotId	ConnectorType	BookingDate	SlotStartTime	SlotLength	EstCost50KWCharger	Bool
3609	mostly68-5cda-4802-ac42-147fd8084d2d	162.0	mostlyaa-93d1-4ed2-b515-864050624e21	214539	CCS	2023-06-01	21	3	55.0	
1902	mostly4f-5877-4179-a9a8-e971da7ed7ff	9.0	mostlya2-db7e-4114-8569-1a85f7b00ac7	567648	CCS	2023-05-01	15	2	55.0	
3066	mostlyd9-f5d0-49ab-a526-fac44398ebfc	2.0	mostly00-2d92-49e6-8d26-c2da4cd41640	234781	CCS	NaT	17	2	55.0	

```
=== Stations ===
```

	Id	ChargingPoints	Latitude	Longitude	OperatorID	Address	
15	mostly20-60a1-4576-a0f0-c0f3915b108b	3	-37.75771	144.955264	TBD		
83	mostlyd1-a726-4d9c-a3b6-d3b0079c8954	5	-37.802159	144.970827	TBD	144RmeTBDwen77intoobour.AmwerctionRdR37catio...	st50uitioceryoChargingPoints
61	mostly22-17aa-4ea7-8995-26ec87db19bc	2	-37.783702	144.89731	TBD	E39StericUChargerHr66sPrPullRestrooms WJ MelU...	

```
=== UserPayment ===
```

	Id	BookingDeposit	PurchaseId	PaymentType	CustPayAmount	TransAmount	UserId	StationId	PaymentDate	Charg
1178	mostlya0-bd1b-4800-953b-e35fc784ff73	1	7.0	Cash	27.957449	35.685663	7.0	0.0	NaT	mos 09bb- ee7a0fb
1797	mostly28-f1e4-4094-833a-13b71269d386	1	94.0	MasterCard	32.84096	27.127117	6.0	14.0	NaT	mos aee5- c588e61
1225	mostlyd4-c6b0-44a1-a5a0-87c5f9937333	1	6.0	Cash	50.028035	30.306651	91.0	9.0	NaT	mos 0c49- b54c5e8

```
=== ChargingSession ===
```

	Id	SlotID	UserId	StationId	LocationId	ChargeStartTime	ChargeEndTime	ChargeTimeHrs	KwhTotal	CustPricePe
2651	mostlyfb-ef95-4d26-9343-20d0754ea3ac	480509	-4454.0	1.0	181553	NaT	NaT	0 days 01:00:00	53.039271	0.5828
1064	mostlyf5-32d6-499b-9eae-0aa2c80c6048	176687	57.0	22.0	698346	2023-05-10 10:12:00	NaT	0 days 01:00:00	76.275731	0.6923

3250	27b1-479f-ae9a-bb9aee69c601	664572	13.0	19.0	492042	NaT	NaT	0 days 01:17:00	55.518466	0.532

```
=== PurchaseWholesaleElec ===
```

	Id	StationId	SupplyId	ExpenditureCode	RRPperKW	RRPperKW24hrs	BasePrice	PeakPriceAdjustment	CustPriceP
822	mostly39-d334-43de-bfc5-1111088c7afc	mostlydd-170c-4b32-ba59-d18c3abb32c2	VIC	Wholesale_cost	0.22699	0.061748	0.55	0	0.57
515	mostly43-7b7a-42f0-a221-2b29311a8dfc	mostlya2-643e-4b71-b2a5-649e3b150469	VIC	Wholesale_cost	-0.05388	-0.041298	0.55	0	0.68
1563	mostlyd9-cc5e-4066-ad1f-1749587ce654	mostly11-3c2c-4b7c-a567-7021089997cb	VIC	Wholesale_cost	0.127252	0.02317	0.55	0	0.69

```
=== ev ===
```

	Id	Manufacturer	Model	VehicleYear	VehicleClass	FuelType	ConnectorType	BatteryCapacityKwh	EstBattChargeTi
69	mostly4f-242b-49e9-b593-1b6b2e803ac5	_RARE_	Tesla Model Y	2024	Medium SUV	BEV	CCS	94.977	
17	mostly10-e119-4a18-84e9-35de448044a6	BMW	_RARE_	2024	Small Car	BEV	CCS	88.524	
49	mostly2e-f3da-4fa2-bbc3-5626f934c0ae	BYD	MG MG4	2024	Small Car	BEV	CCS	81.487	

Comparison For Charging Session

```
# original Charging table data
ChargingSession.head(2)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

	Id	SlotID	UserId	StationId	LocationId	ChargeStartTime	ChargeEndTime	ChargeTimeHrs	KwhTotal	CustPricePerKW	Cu
0	4926737	841344	78908148	730023	878393	7/10/2023 7:18	7/10/2023 8:44	0 days 01:26:00	82.049511	0.570110	
1	3738844	252642	78908148	730023	878393	5/08/2023 5:29	5/08/2023 7:01	0 days 01:32:00	88.246562	0.532488	

Next steps: Generate code with ChargingSession | View recommended plots | New interactive sheet

Generate: copy ChargingSession to dataframe Original_charging | Close

1 of 1 | Use code with caution

```
# prompt: copy ChargingSession to dataframe Original_charging

Original_charging = ChargingSession.copy()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

```
synthetics['ChargingSession'].head(2)
```

```
rnel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
```

ChargeStartTime	ChargeEndTime	ChargeTimeHrs	KwhTotal	CustPricePerKW
NaT	NaT	_RARE_	70.019863	0.586903
2023-11-04 17:43:00	2023-07-04 18:46:00	_RARE_	47.591555	0.65694

Generate

copy synthetics['ChargingSession'] to dataframe Synthetics_charging

Close

1 of 1

Use code with caution

```
# prompt: copy synthetics['ChargingSession'] to dataframe Synthetics_charging

Synthetics_charging = synthetics['ChargingSession'].copy()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

Generate

Create a plot to compare the length of both dataframe Original_charging and Synthetics_charging

Close

1 of 1

Use code with caution

```
# prompt: Create a plot to compare the length of both dataframe Original_charging and Synthetics_charging

import matplotlib.pyplot as plt

# Assuming 'Original_charging' and 'Synthetics_charging' are already defined DataFrames.

# Calculate the lengths of the DataFrames.
original_length = len(Original_charging)
synthetic_length = len(Synthetics_charging)

# Create a bar plot to compare the lengths.
plt.figure(figsize=(8, 6))
plt.bar(['Original Charging', 'Synthetic Charging'], [original_length, synthetic_length], color=['skyblue', 'lightcoral'])
plt.ylabel('Number of Records')
plt.title('Comparison of DataFrame Lengths')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```

Comparison of DataFrame Lengths

Generate

Create a plot to compare the number of unique value in columns UserId, StationId, for both dataframe Original_charging and Synthetics_charging

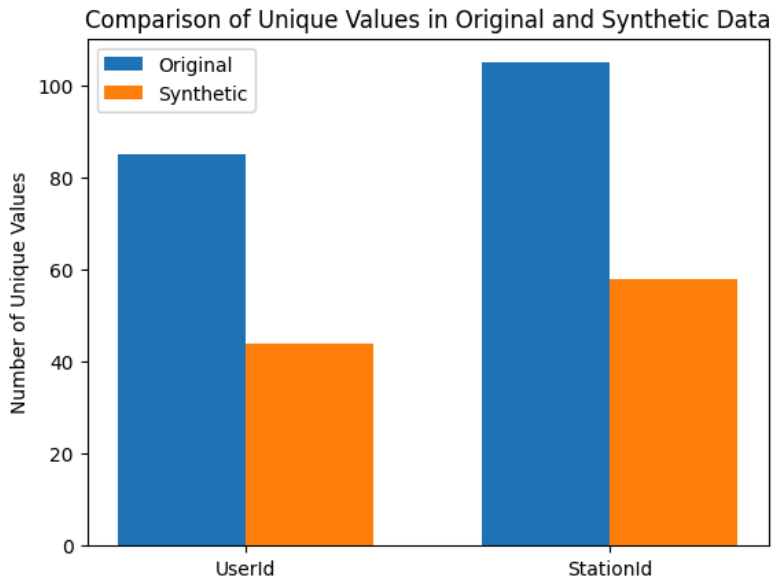
Close

1 of 1

Use code with caution

```
# prompt: Create a plot to compare the number of unique value in columns UserId, StationId, for both dataframe Original_charging and

import matplotlib.pyplot as plt

# Calculate the number of unique values for UserId and StationId in both DataFrames
unique_original_userid = Original_charging['UserId'].nunique()
unique_original_stationid = Original_charging['StationId'].nunique()

unique_synthetic_userid = Synthetics_charging['UserId'].nunique()
unique_synthetic_stationid = Synthetics_charging['StationId'].nunique()

# Create a bar plot for comparison
columns = ['UserId', 'StationId']
original_values = [unique_original_userid, unique_original_stationid]
synthetic_values = [unique_synthetic_userid, unique_synthetic_stationid]

x = range(len(columns))  # the label locations
width = 0.35  # the width of the bars

fig, ax = plt.subplots()
rects1 = ax.bar(x, original_values, width, label='Original')
rects2 = ax.bar([i + width for i in x], synthetic_values, width, label='Synthetic')

# Add some text for labels, title and custom x-axis tick labels, etc.
ax.set_ylabel('Number of Unique Values')
ax.set_title('Comparison of Unique Values in Original and Synthetic Data')
ax.set_xticks([i + width / 2 for i in x])
ax.set_xticklabels(columns)
ax.legend()

plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```



Generate

Plot the sns.hist Distribution of Energy Consumed kwhTotal for both dataframe Original_charging and Synthetics_charging

Close

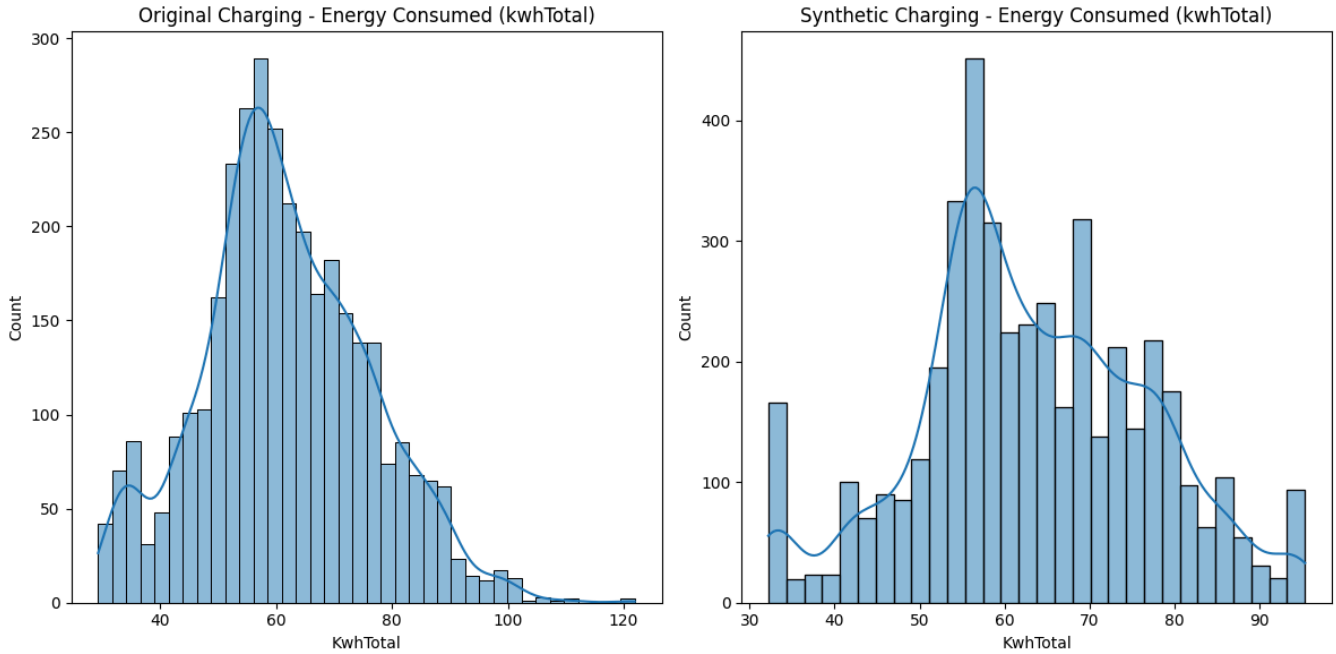
1 of 1

Use code with caution

```
# prompt: Plot the sns.hist Distribution of Energy Consumed kwhTotal for both dataframe Original_charging and Synthetics_charging

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
```

```
plt.subplot(1, 2, 1)
sns.histplot(Original_charging['KwhTotal'], kde=True)
plt.title('Original Charging - Energy Consumed (kwhTotal)')

plt.subplot(1, 2, 2)
sns.histplot(Synthetics_charging['KwhTotal'], kde=True)
plt.title('Synthetic Charging - Energy Consumed (kwhTotal)')

plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283: DeprecationWarning: `should_run_async` will not call `transfo
  and should_run_async(code)
```



Generate

Plot the sns.hist Distribution of chargeTimeHrs for both dataframe Original_charging and Synthetics_charging

Close

1 of 1

Use code with caution

```
# prompt: Plot the sns.hist Distribution of chargeTimeHrs for both dataframe Original_charging and Synthetics_charging

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.histplot(Original_charging['ChargeTimeHrs'], kde=True)
plt.title('Original Charging - Charge Time (hrs)')

plt.subplot(1, 2, 2)
sns.histplot(Synthetics_charging['ChargeTimeHrs'], kde=True)
plt.title('Synthetic Charging - Charge Time (hrs)')

plt.tight_layout()
plt.show()
```